JAVA SE

Quiz yourself: The Java Platform Module System and the ServiceLoader class

Yes, the JPMS ServiceLoader class loads services. How's your knowledge of the details?

by Simon Roberts and Mikalai Zaikin

August 16, 2021

If you have worked on our quiz questions in the past, you know none of them is easy. They model the difficult questions from certification examinations. We write questions for the certification exams, and we intend that the same rules apply: Take words at their face value and trust that the questions are not intended to deceive you but to straightforwardly test your knowledge of the ins and outs of the language.

Which statement about the Java Platform Module System (JPMS) is true? Choose one.

A. The `ServiceLoader.load(Class service)` returns an `Iterator` instance.

The answer is A.

B. The `ServiceLoader.load(Class service)` returns a `ServiceLoader` instance.

The answer is B.

C. JPMS introduces a mechanism to load service implementations from the `META-INF/service` directory.

The answer is C.

D. Changing the implementation of a service requires recompiling the module definition of the client module.

The answer is D.

E. A service provider class must implement an interface that defines the API of the service.

The answer is E.

F. A service provider class must have a public zero-argument constructor.

The answer is F.

Answer. Option A suggests that the return type of the `load` method is an `Iterator`. In fact, all three `load(...)` overloaded methods return parameterized `ServiceLoader` instances. The `ServiceLoader` class does implement `Iterable` (and therefore it can be used directly in the enhanced `for` construction); however, the class does not implement the `Iterator` interface. From this you can determine that option A is incorrect and that option B is correct.

Option C suggests a *new* mechanism that involves listing class names for service providers in the `META-INF/service` directories of JAR files. In fact, such a mechanism existed prior to Java 9, and JAR files containing such services would be placed on `classpath`. However, Java 9 introduced a new mechanism such that a service loader can look up module definitions on the *module path* and collect service implementations. Since the mechanism alluded to is neither new nor specific to JPMS, option C is incorrect.

Option D is incorrect as well. A powerful benefit of the service loading mechanism in JPMS is that no recompilation is required. Simply adjusting the module path or the modules placed on that path as the application starts up is sufficient to change the set of service implementations available.

Option E suggests that a service provider must implement an *interface*. This is not accurate; the service *can* be described using an interface, but a class, either concrete or abstract, is fine too. (Note that there are some constraints and more options not mentioned here.) From this you know that option E is incorrect.

Option F suggests that the implementation class must have a zero-argument constructor. This is certainly common, but it's not *mandatory*. Instead the service implementation class can define a `public static` zero-argument method `provider` that is used as a factory for the service implementation object. If a `provider()` method does not exist, the service loader will fall back to building the service object using the zero-argument constructor. From this you know that option F is incorrect.

Conclusion. The correct answer is option B.

Related quizzes

- Quiz yourself: Migrating to the Java Platform Module System
- Quiz yourself: Declaring and accessing modules
- Quiz yourself: Module definitions and automatic modules

Simon Roberts

Simon Roberts joined Sun Microsystems in time to teach Sun's first Java classes in the UK. He created the Sun Certified Java Programmer and Sun Certified Java Developer exams. He wrote several Java certification guides and is currently a freelance educator who publishes recorded and live video training through Pearson InformIT (available direct and through the O'Reilly Safari Books Online service). He remains involved with Oracle's Java certification projects.

Mikalai Zaikin

Mikalai Zaikin is a lead Java developer at IBA IT Park in Minsk, Belarus. During his career, he has helped Oracle with development of Java certification exams, and he has been a technical reviewer of several Java certification books, including three editions of the famous *Sun Certified Programmer for Java* study guides by Kathy Sierra and Bert Bates.

Share this Page